LogiComp 301

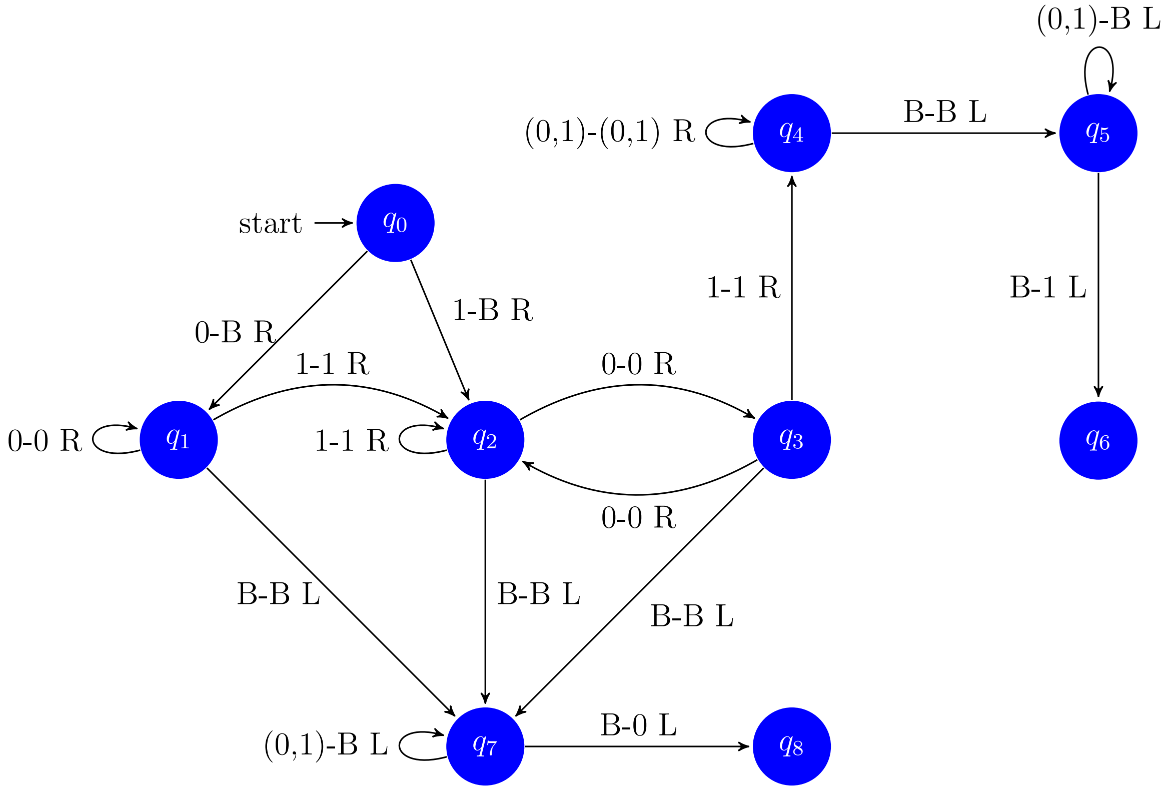
Assignment One

Steven Kerr 6022796

04/04/2019

1.(i) Turning machine for $L = \{s \in \{0,1\}^* |$ s contains at least one substring of the form $1v1$ where v contains only 0s and where the length of v is odd$\}$

TM M



A formal definition of the above Turing machine

δ	0	1	B
q_0	q_1	q_2	-
q_1	q_1	q_2	q_7
q_2	q_3	q_1	q_7
q_3	q_2	q_4	q_7
q_4	q_4	q_4	q_5
q_5	q_5	q_5	q_6
q_6	*Accepting*	-	*State*
q_7	q_7	q_7	q_8
q_8	*Rejecting*	-	*State*

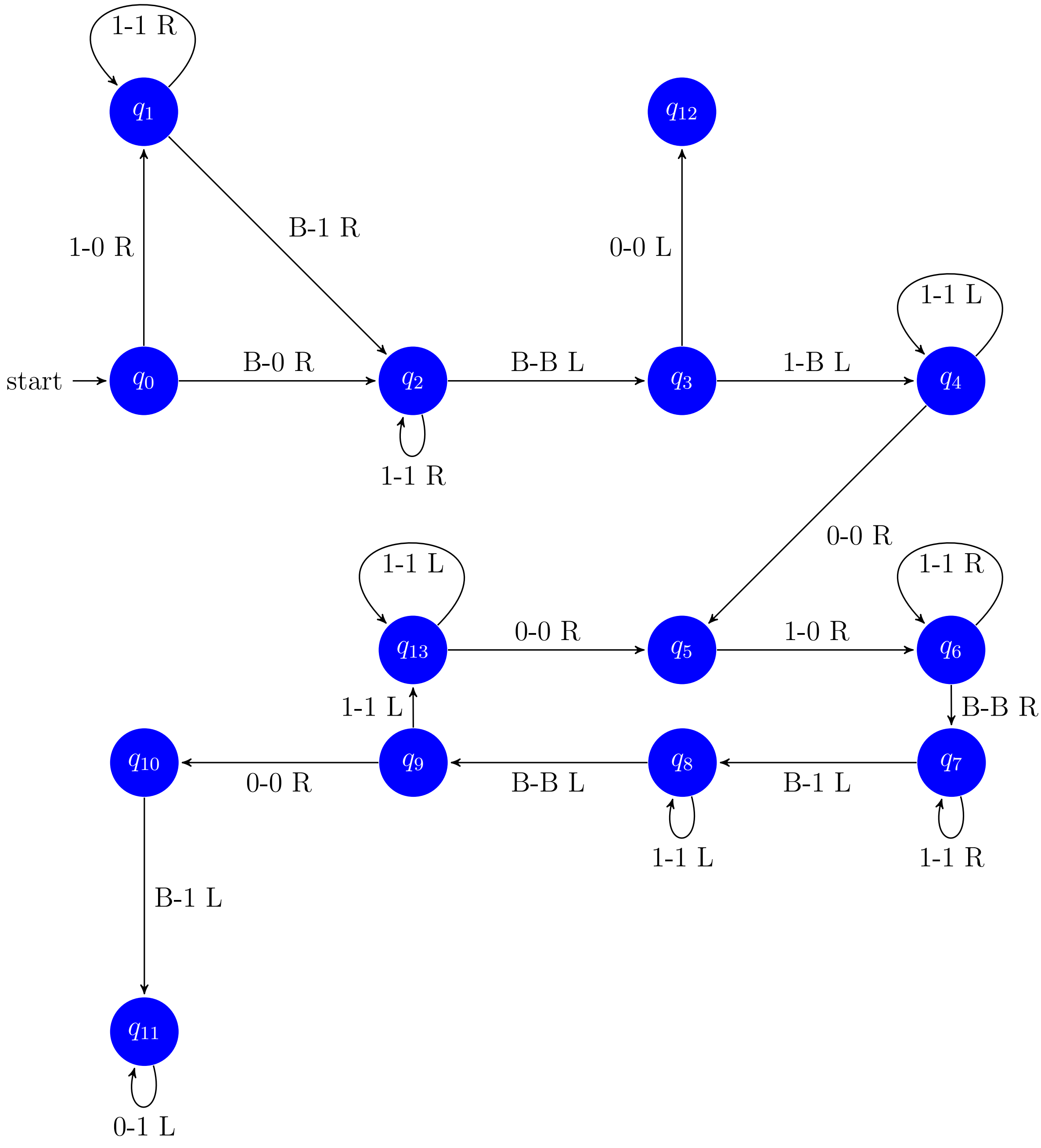
- $Q = \{q_0, q_1, \ldots, q_8\}$
- $\Sigma = \{0, 1\}^*$
- q_0 is the start state
- q_6 is the "Accepting state"
- q_8 is the "Rejecting state"

description:
q_0 marks the beginning of the tape with a B. $q_1,\ q_2\ and\ q_3$ decide if there is a string $1v1$ string such that v contains only $0's$ and $|v|$ is odd length. note, $q_1 \to q_2$ can only be achieved if a string starts with 0^n and eventually a 1 is found, otherwise the string is rejected to q_7. When in q_2 a 1 has been found and loops on 1 until a 0 is found. If there are odd number of $0's$ the head will be in q_2 and either a 1 will be found $q_3 \to q_4$ and the string will be accepted and sent to q_4, or a 1 will be found $q_2 \to q_2$ and the loop will start again, or a B will be found in $q_1,\ q_2\ or\ q_3$ and the string will be rejected to q_7.

(ii) Computation for above machine on string 01010:
$q_0 01010 \to 0q_1 1010 \to 01q_2 010 \to 010q_3 10 \to 0101q_4 0 \to 01010q_4 \to 0101q_5 0 \to 010q_5 1B \to 01q_5 0BB \to 01q_5 0BBB \to 0q_5 1BBBB \to q_5 0BBBBB \to q_5 BBBBBB \to q_6 1BBBBB$

2. Turing machine for $f(x, y) = 2(x + y)$

TM M



A formal definition of the above Turing machine

- $Q = \{q_0, q_1, \ldots, q_13\}$
- $\Sigma = \{0, 1\}^*$
- q_0 is the start state
- q_{11} is the "Accepting state"
- q_{12} is the "Rejecting state"

δ	0	1	B
q_0	-	q_1	q_2
q_1	-	q_1	q_2
q_2	-	q_2	q_3
q_3	q_{12}	q_4	-
q_4	q_5	q_4	-
q_5	-	q_6	-
q_6	-	q_6	q_7
q_7	-	q_7	q_8
q_8	-	q_8	q_9
q_9	q_{10}	q_{13}	-
q_{10}	-	-	q_{11}
q_{11}	q_{11}	-	-
q_{12}	*Rejecting*	-	*State*
q_{13}	q_5	q_{13}	-

Description:
q_0 is the starting state and marks the beginning of the tape with a 0. It also determines if $x = 0$ (assuming $x = 0, y = 2$ will look like $\{B11BB\ldots\}$). If $x > 0$ then q_1 loops right until a B is read. Once a B is read it is changed to a 1 and moves to q_2. However, if $x = 0$ then q_0 changes the B into a 0 and moves straight to q_2. q_2 deals with y. If $y > 0$ then q_2 loops on itself moving right on the tape until a B is found. Otherwise, if $y = 0$ the head moves left to q_3. If the head is looking at a 0 this means that $x + y = 0$ and the machine moves to q_{12} and stops with the output 0. When combining $x + y$ a 1 is added where the B separates $x\,and\,y$. $q_3 \to q_4$ removes the 1 added, then $q_4 \to q_5$ moves the head back to the beginning of the tape and prepares for multiplication by two. The loop from $q_5 \to q_6 \to q_7 \to q_8 \to q_9 \to q_{13} \to q_5$ works in the following way;
When the loop is entered, the computation will look like so, $0q_5 11\ldots 1^n BB\ldots$. q_5 will change the 1 to a 0 and move right till a blank is found. Once a blank is found, it will be left there to mark the center ($q_6 \to q_7$) and the machine will keep moving right over the $1's$ until another blank is found. for example: $011\ldots 1^n Bq_7 BBB \cdots \to 011\ldots 1^n q_8 B1BB\ldots$. q_8 will keep moving left over all the $1's$ until a blank is found (the marker for the center), then the machine continues to move left over all the $1's$ ($q_9 \to q_{13} \to q_5$) and the loop starts again.
If a 0 is found in state q_9, this means that the multiplication loop is over, example: $000\ldots q_9 0^n B111\ldots 1^{n-1}$, $q_{10} \to q_{11}$ changes the B (center marker) to a 1 and finally q_{11} loops left, changing all the $0's$ to $1's$.

3.(a)$A\&B$ are infinite in size. Also, $N \approx A$ and $N \approx B$ where N is the set of natural numbers.

(i) $A \cup B$ is countably infinite: yes
We know that $A\&B$ are both countably infinite. Let $A = \{a_0, a_1, \ldots, a_i \ldots\}$ and $B = \{b_0, b_1, \ldots, b_i \ldots\}$ using the following function:

$$f(n) = \begin{cases} b_{n/2} & \text{if } n \text{ is even} \\ a_{\frac{n+1}{2}} & \text{if } n \text{ is odd} \end{cases}$$

We can now see that $f(n)$ has a bijection on the natural numbers. This means that $A \cup B$ also has a bijection on the natural numbers and by the definition of "countably infinite" $A \cup B$ remains countably infinite.

(ii) $A \cap B$ is countably infinite: yes
Since we know A & B are countably infinite we know $A \cap B \subseteq A$ and A is countably infinite. To be more specific, Let $A = \{a_0, a_1, \ldots, a_i \ldots\}$ and $B = \{b_0, b_1, \ldots, b_i \ldots\}$. We can create a bijection between $N \approx A \cap B$ such that $A \cap B = \{a_0 \cap b_0, a_1 \cap b_1, \ldots, a_i \cap b_i \ldots\}$. Hence, by the definition of "countably infinite" $A \cap B$ remains countably infinite.

(iii) $A - B$ is countably infinite, where $\{x|x \in A \,\&\, x \notin B\}$: depends

1. If $A = B$ then $A - B = \emptyset$ which is finitely countable.
2. If $\{\forall a \in A \mid a \notin B)\}$ then A remains unchanged and is still countably infinite.
3. If $\{\exists a \in A \mid a \in B\}$ then we know that $A - B \neq \emptyset$ but we don't know if it is still infinity large or finitely large. We do however still know that it is countable because we can still have a bijection to the natural numbers. Therefore, $A - B$ is either finitely countable or infinitely countable.

3. (b) Let F be the set of all total unary functions $f : N \to N$ and F_{TC} be the set of all total unary Turing-computable functions $f : N \to N$.
Let f_i^1 be the unary function from N into N computed by Turing machine M_i. Now, let F_i^k be the k-ary function from N^k into N computed by machine M_i. We can now see that for each $k \geq 1$ the set of $F_{TC} = \{f_1^k, f_2^k, f_3^k, \dots\}$ which is a countably infinite set.
We can also see that the set $F_{TC} \subseteq F$ which implies that F is also countably infinite. If we took away the set F_{TC} from the set F $(F_{TC} - F)$ then we would be left with the set of all total unary functions that are not Turing-computable. This is also a sub set of F and therefore is also countably infinite.

4. Does Godel's incompleteness theorem imply the impossibility of Hilbert's programme?

Hilbert's program was designed to provide a secure foundation for all mathematics. This proposition included a requirement for all mathematical statements to be written in precise formal language and manipulated according to well defined rules (formal system). Completeness where a proof that all true mathematical statements can be proved in the adopted formal system. Consistency where a proof has no contradictions that can be obtained in the formal system. Decidability where an algorithm can be constructed for deciding, in the formal system, the truth or falsity of any mathematical statement. Finally, Conservation where a proof that any result about "real objects" obtained using "ideal objects" can be restated without using ideal objects.

Godel's incompleteness theorem shows that there is a gap between "truth" and "proof" where there are some true statements which can not be proved within any mathematical system. Axioms are an important part of Godel's incompleteness theory. If we do not have all of the axioms to prove a true statement we can just add that as a new axiom and it will expand what we can prove within mathematics. This is very important for Godel because we are trying to prove that there will be a set of axioms from which we can deduce all truths of mathematics. Godel produced what is known as Godel coding. Godel devised a way to code all mathematical statements, including whether they are true or not. This gave him the ability to compare statements with one another and deduce a proof.

A simple example of the paradox that Godel's incompleteness theorem shows is this. Imagine the statement "This statement cannot be proven mathematically" this statement is obviously either true or false. So, lets start by assuming the statement is false. That would mean "This statement can be proven mathematically" is true, but a provable statement must be true, so we have started with something we assumed was false and now we have deduced that it is true. This is a contradiction. This is a problem for Hilbert's programme because consistency requires a proof to have no contradictions. This means our statement "This statement cannot be proven mathematically" can not be false, so it must be True. But now we have a statement "This statement cannot be proven mathematically" is true, but can not be proven. Therefore, we have found a true statement which can not be proven true within that system. This is another problem for Hilbert's programme because completeness states that a true statement can be proven.

Godel's theorem goes on to explain that we could add the axiom to make the above truth statement provable in our system, but that will not help as no matter what, a new statement can always be shown true but not provable in our new system. As I understand it, no matter how much you expand mathematics by adding axioms, the system will always be missing something. This would make it impossible to create a formal system of mathematics and again adding to the impossibility of Hilbert's programme.

In conclusion, Godel's incompleteness theorem does imply the impossibility of Hilbert's programme ever being satisfied.